

IN THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Cancelled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (cancelled), (withdrawn), (new), (previously presented), or (not entered).

Please AMEND claims 1, 7 and 13 in accordance with the following:

1. (CURRENTLY AMENDED) A compiler embodied on a medium for compiling a source program, said compiler comprising:

detecting a parallelization directive described by a user in said source program; and

if said parallelization directive is detected, generating a front-end intermediate language for said parallelization directive by positioning on a storage region, each processing code of at least part of the parallelization directive with a hierarchical structure in accordance with an internal structure of said parallelization directive.

2. (PREVIOUSLY PRESENTED) The compiler according to claim 1, further comprising:

adding to said front-end intermediate language of a statement to which the parallelization directive is applied, reference information from said front-end intermediate language of said statement to which the parallelization directive is applied, to said front-end intermediate language for the parallelization directive.

3. (PREVIOUSLY PRESENTED) The compiler according to claim 1, further comprising:

by using a processing table which stores one or a plurality of items of processing information for each of said processing codes, acquiring the processing information corresponding to a current processing content based on said processing code within the front-end intermediate language for said parallelization directive.

4. (PREVIOUSLY PRESENTED) The compiler according to claim 3, wherein said current processing content is one of type analysis, syntactic analysis, semantic analysis, and generation of a compiler intermediate language.

5. (PREVIOUSLY PRESENTED) The compiler according to claim 1, wherein said hierarchical structure is a list structure.

6. (PREVIOUSLY PRESENTED) The compiler according to claim 1, wherein said part of said parallelization directive comprises a directive, a clause, and a line, and

a processing code for said directive is linked downward to a processing code for said clause, and

said processing code for said clause is linked downward to a processing code for said lines.

7. (CURRENTLY AMENDED) A compiling method for compiling a source program, said compiling method comprising:

detecting a parallelization directive described by a user in said source program; and

if said parallelization directive is detected, generating a front-end intermediate language for said parallelization directive by positioning on a storage region, each processing code of at least part of the parallelization directive with a hierarchical structure in accordance with an internal structure of said parallelization directive.

8. (PREVIOUSLY PRESENTED) The compiling method according to claim 7, further comprising:

adding to said front-end intermediate language of a statement to which the parallelization directive is applied, reference information from said front-end intermediate language of said statement to which the parallelization directive is applied, to said front-end intermediate language for the parallelization directive.

9. (PREVIOUSLY PRESENTED) The compiling method according to claim 7, further comprising:

by using a processing table which stores one or a plurality of items of processing information for each of said processing codes, acquiring the processing information corresponding to a current processing content based on said processing code within the front-end intermediate language for said parallelization directive.

10. (ORIGINAL) The compiling method according to claim 9, wherein said current processing content is one of type analysis, syntactic analysis, semantic analysis, and generation of a compiler intermediate language.

11. (ORIGINAL) The compiling method according to claim 7, wherein said hierarchical structure is a list structure.

12. (ORIGINAL) The compiling method according to claim 7, wherein said part of said parallelization directive comprises a directive, a clause, and a line, and a processing code for said directive is linked downward to a processing code for said clause, and

 said processing code for said clause is linked downward to a processing code for said lines.

13. (CURRENTLY AMENDED) A compiling apparatus for compiling a source program, comprising:

 a detector that detects a parallelization directive described by a user in said source program; and

 a generator that generates a front-end intermediate language for said parallelization directive by positioning on a storage region, each processing code of at least part of the parallelization directive with a hierarchical structure in accordance with an internal structure of said parallelization directive if said parallelization directive is detected.

14. (PREVIOUSLY PRESENTED) The compiling apparatus according to claim 13, further comprising:

 an adder that adds to said front-end intermediate language of a statement to which the parallelization directive is applied, reference information from said front-end intermediate language of a statement to which the parallelization directive is applied, to said front-end intermediate language for the parallelization directive.

15. (PREVIOUSLY PRESENTED) The compiling apparatus according to claim 13, further comprising:

 an acquiring unit that acquires, by using a processing table which stores one or a

plurality of items of processing information for each of said processing codes, the processing information corresponding to a current processing content based on said processing code within the front-end intermediate language for said parallelization directive.

16. (ORIGINAL) The compiling apparatus according to claim 15, wherein said current processing content is one of type analysis, syntactic analysis, semantic analysis, and generation of a compiler intermediate language.

17. (ORIGINAL) The compiling apparatus according to claim 13, wherein said hierarchical structure is a list structure.

18. (ORIGINAL) The compiling apparatus according to claim 13, wherein said part of said parallelization directive comprises a directive, a clause, and a line, and a processing code for said directive is linked downward to a processing code for said clause, and

 said processing code for said clause is linked downward to a processing code for said lines.